Detailed Action

1. This is the initial office action bases on the application filed on February, 21st, 2007 which claims 16 to 30 are presented for examination.

Status of Claims

2. Claims 16-30 are pending, of which claims 16, 27, 28 and 29 are in independent form.

Oath/Declaration

3. The office acknowledges receipt of a properly signed oath/declaration filed on February, 21st, 2007.

Effective filing date

4. The effective filing date that has been considered for this application is May 5th, 2006.

Information Disclosure Statement

5. The information disclosure statements filed on May 5th, 2006 comply with the provisions of 37 CFR 1.97, 1.98. They have been placed in the application file and the information referred to therein has been considered as to the merits.

Claim Rejections - 35 USC § 101

6. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful

> improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

7. Claims 27-30 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

As per claim 27, wherein claim 27 recites "A device for stimulating for controlling operating sequences, the function accessing at least one global variable of at least one program for control" – even though the limitation recites the word "device"; however, claim itself does not disclose any hardware component to realize any of the underlying functionality of the said device's implementation. Therefore, examiner will reasonably interpret the functionality of this device as an implementation of software alone, Software and computer program *per se* do not fall within any category of patent-eligible subject matter. Claim 27 is therefore ineligible for patent protection. *See* MPEP § 2106.

As per claim 28, wherein claim 28 recites "A control unit, comprising: a device for stimulating functions for control of operating sequences, the functions accessing at least one global variable of at least one program for control" –as pointed out above the recited phrase "A control unit" is not followed by any disclosed hardware component in the claim to realize any of the hardware implemented functionality of the said control unit. Therefore, examiner will reasonably interpret functionality of this control unit as an implementation of software alone, Software and computer program *per se* do not fall within any category of patent-eligible subject matter. Claim 28 is therefore ineligible for patent protection. *See* MPEP § 2106.

As per claim 29 and claim 30, claims 29 and 30 recite "A computer program product for stimulating functions for controlling operating sequences", and claim 30 further recites "wherein a code of the product is executed on a computer". According to the recited claims, it is clearly evident that the inventive claims are directed towards executable software code, not towards statutory tangible product such as a non-transitory computer readable medium. Software and computer program *per se* do not fall within any category of patent-eligible subject matter, and accordingly as set forth in 35 U.S.C. § 101, claims 29 and 30 are therefore ineligible for patent protection. *See* MPEP § 2106.

Double Patenting

8. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

9. Claim 16-26 are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over Claim(s) 1, 2,4, 5, 6, and 7 of US Patent 7593791 B2. Although the conflicting claims are not identical, they are not patentably distinct from each other because they are drawn to obvious variations.

<u>Per claim 16:</u>

Claim 16 of the instant application is recited as follows:

A method for stimulating functions for controlling operating sequences, the functions accessing at least one global variable of at least one program for control, the method comprising:

Claim 1 of the Patent (Patent No 7,593,791 B2), lines 1-4, recites as follows:

(A method for adaptation of a function for controlling an operating sequence of a control unit, the function accessing at least one global variable of at least one program for control, the method comprising:)

providing at least one stimulation function;

Claim 1 of the Patent (Patent No 7,593,791 B2), lines 1-4, recites as follows:

(A method for adaptation of a function for controlling an operating sequence of a control unit, the function accessing at least one global variable of at least one program for control)

Thus, it would have been obvious to one ordinary skill in the art to realize that either the stimulation function or the adaptation function can be utilized to control a sequence of operations in a control unit because in order to control an operating sequence of a control unit an executable code function access the global variable of at least one of the said control function and replace the global variable value with the address references of the function.

and

accessing the at least one global variable via at least one software breakpoint

Claim 1 of the Patent (Patent No 7,593,791 B2), line 15, recites as follows: (determining a breakpoint in the at least one program); and

Claim 1 of the Patent (Patent No 7,593,791 B2), lines 11 and 12, recites as follows:

(assigning the global variable address information which is present in at least one memory device)

Thus it would have been obvious to one ordinary skill in the art to recognize that after determining the break point of the function at first then the global variable address information is accessed to assign and replace address information –*See* remaining limitations of claim 1 of the patent.

Per claim 17:

The method as recited in Claim 16, wherein the functions are stimulated within the at least one program during a runtime of the at least one program.

Due to the similarities of both the instant application and the identified Patent (Patent No 7,593,791 B2), it is therefore obvious that one of ordinary skill in the art would propagate or execute the external function after loading the global variable for execution of the original controlling function because the patent (Patent No 7,593,791 B2) discloses –address information of the global variables being loaded out of the memory device by at least one load instruction and this address information of the global variable of the load instruction being advantageously replaced (Col 2:36-40), which means functions (original) are adapted (by external function) during runtime of the sequential controlling program.

Per claim 18:

The method as recited in Claim 16, wherein the functions are stimulated in real time within a runtime system.

Due to the similarities of both the instant application and the identified Patent (Patent No 7,593,791 B2), it is therefore obvious that one of ordinary skill in the art would propagate or execute the external function after loading the global variable for execution of the original controlling function because the patent (Patent No 7,593,791 B2) discloses –address information of the global variables being loaded out of the memory device by at least one load instruction and this address information of the global variable of the load instruction being advantageously replaced (Col 2:36-40), which means functions (original) are adapted (by external function) during runtime of the sequential controlling program.

Per claim 19:

assigning the at least one global variable a first piece of data information;
replacing the first piece of data information by a second piece of data information corresponding to new stimulation values

Claim 1 of the Patent (Patent No 7,593,791 B2), lines 11- 16, recites as follows: (assigning the global variable address information which is present in at least one memory device; loading the memory device the address information of the global variable by at least one load instruction; and replacing the address information of the global variable (first piece of data information) with one of the store instruction and the load instruction (second piece of data information)).

Per claim 20:

The method as recited in Claim 16, further comprising:
stimulating the functions by an internal bypass
Claim 1 of the Patent (Patent No 7,593,791 B2), lines 1-4, recites as follows:
(A method for adaptation of a function for controlling an operating sequence of a control unit, the function accessing at least one global variable of at least one program for control).
Thus, - Although the functions may be adapted using an external bypass, the advantageous embodiment is to perform the adaptations internally in such a way that they are tied into the program run and therefore there are dynamic hooks for software interventions without any source code changes.

In accordance with an example embodiment, the address information of load instructions is modified, the content of store instructions is modified, the address information of function is modified and new program codes are added. These changes are implemented here in the exemplary embodiment on an existing software program version on the basis of targeted hex code modifications –See Patent 7593791, specification Col4:10-21.

Per claim 21:
The method as recited in Claim 16, wherein:
the software breakpoint is accomplished by assigning address information to the at least one global variable, the address information is loaded from a memory means by a load instruction, and *the address information of the at least one global variable of the load*
instruction is replaced.
Claim 1 of the Patent (Patent No 7,593,791 B2), lines 5, and 11-16 recites as follows:
(determining a breakpoint in the at least one program;
assigning the global variable address information which is present in at least one memory device; loading the memory device the address information of the

global variable by at least one load instruction; and replacing the address information of the global variable with one of the store instruction and the load instruction).

<u>Per claim 22:</u>

The method as recited in Claim 21, further comprising:
replacing the address information of the at least one global variable by address information of a pointer variable.
Claim 2 of the Patent (Patent No 7,593,791 B2), lines 1-3 recites as follows:
(The method as recited in claim 1, wherein the address information of the global variable is replaced by address information of a pointer variable).

<u>Per claim 23:</u>

The method as recited in Claim 21, further comprising:
determining an initial address of the at least one stimulation function from the address information
Claim 5 of the Patent (Patent No 7,593,791 B2), lines 1-3 recites as follows:
(determining an initial address of the function from the address information).

<u>Per claim 24:</u>

The method as recited in Claim 21, further comprising:
replacing functions for control of operating sequences by replacing the address information by additional functions
Claim 6 of the Patent (Patent No 7,593,791 B2), lines 2-4 recites as follows:
(replacing the function for controlling the operating sequence by replacing the address information with an additional function).

<u>Per claim 25:</u>

The method as recited in Claim 16, wherein:
the software breakpoint is accomplished by addressing the at least one global variable via a store instruction,
Claim 1 of the Patent (Patent No 7,593,791 B2), lines 5, and 15-16 recites as follows:
(determining a breakpoint in the at least one program
replacing the address information of the global variable with one of the store instruction)
and
the store instruction is manipulated onto the at least one global variable by replacing the store instruction by a jump instruction
Claim 4 of the Patent (Patent No 7,593,791 B2), lines 2-4 recites as follows:
(manipulating a memory instruction onto the global variable by replacing the memory instruction with a jump instruction).

<u>Per claim 26:</u>
The method as recited in Claim 25, *wherein the functions for controlling the operating sequences by replacing the store instruction by the jump instruction are replaced by additional functions*
Claim 7 of the Patent (Patent No 7,593,791 B2), lines 1-4 recites as follows:
(the function for controlling the operating sequence by replacing the memory instruction with the jump instruction is replaced with an additional function).

10. Claim 27 is rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over Claim 8 of US Patent No 7,593,791 B2. Although the conflicting claims are not identical, they are not patentably distinct from each other because they are drawn to obvious variations.

Per claim 27:

A device for stimulating functions for controlling operating sequences, the functions accessing at least one global variable of at least one program for control, comprising:

Claim 8 of the Patent (Patent No 7,593,791 B2), lines 1-5 recites as follows:

(A device for adopting a function for controlling an operating sequence of a control unit, the function accessing at least one global variable of at least one program for control, the global variable being assigned address information, the device comprising)

a first stimulation arrangement for activating at least one stimulation function;

Claim 8 of the Patent (Patent No 7,593,791 B2), lines 1-3 and 7-8 recites as follows:

(A device for adopting a function for controlling an operating sequence of a control unit, the function accessing at least one global variable of at least one program for control

a control device configured to: determine a breakpoint in the at least one program)

Thus, it would have been obvious to one ordinary skill in the art to realize that either the stimulation function or the adaptation function can be utilized to control a sequence of operations in a control unit because in order to control an operating sequence of a control unit an executable code function access the global variable of at least one of the said control function and replace the global variable value with the address references of the function.

and

a control arrangement for generating at least one software breakpoint,

Claim 8 of the Patent (Patent No 7,593,791 B2), lines 7-8 recites as follows:

(a control device configured to: determine a breakpoint in the at least one program) *the at least one stimulation function accessing the at least one global variable via the software breakpoint*

Claim 8 of the Patent (Patent No 7,593,791 B2), lines 7-15 recites as follows:

(a control device configured to: determine a breakpoint in the at least one program, wherein the determination is based on a comparison at least one of (1) between an address of the control unit and a register content loaded by a load instruction and (2) between the address of the control unit and a destination address of a store instruction; assign the global variable address information which is present in at least one memory device)

Thus it would have been obvious to one ordinary skill in the art to recognize that after determining the break point of the function at first then the global variable address information is accessed to assign and replace address information –*See* remaining limitations of claim 8 of the patent.

11. Claim 28 is rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over Claim 10 of US Patent No 7,593,791 B2. Although the conflicting claims are not identical, they are not patentably distinct from each other because they are drawn to obvious variations.

<u>Per claim 28:</u>

A control unit, comprising:
a device for stimulating functions for control of operating sequences, the functions accessing at least one global variable of at least one program for control, the device including:

Claim 10 of the Patent (Patent No 7,593,791 B2), lines 1-7 recites as follows:

(A control unit for controlling an operating sequence, the control unit comprising: a device configured to adopt a function to controlling the operating sequence of a control unit, the function accessing at least one global variable of at least one program for control, the global variable being assigned address information, the device comprising)
a first stimulation arrangement for activating at least one stimulation function,

Claim 10 of the Patent (Patent No 7,593,791 B2), lines 3-7 recites as follows:

(a device configured to adopt a function to controlling the operating sequence of a control unit, the function accessing at least one global variable of at least one program for control),
and
a control arrangement for generating at least one software breakpoint

Claim 10 of the Patent (Patent No 7,593,791 B2), lines 8-10 recites as follows:

(a memory in which the address information is located; a control device configured to: determine a breakpoint in the at least one program),

the at least one stimulation function accessing the at least one global variable via the software breakpoint

Claim 10 of the Patent (Patent No 7,593,791 B2), lines 9-17 recites as follows:

(a control device configured to: determine a breakpoint in the at least one program, wherein the determination is based on a comparison at least one of (1) between an address of the control unit and a register content loaded by a load instruction and (2) between the address of the control unit and a destination address of a store instruction; assign the global variable address information which is present in at least one memory device)

Thus it would have been obvious to one ordinary skill in the art to recognize that after determining the break point of the function at first then the global variable address information is accessed to assign and replace address information –*See* remaining limitations of claim 10 of the patent.

12. Claims 29-30, are rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over Claim(s) 12 of US Patent No 7,593,791 B2. Although the conflicting claims are not identical, they are not patentably distinct from each other because they are drawn to obvious variations.

Per claim 29:

A computer program product for stimulating functions for controlling operating sequences, the functions accessing at least one global variable of at least one program for control, the product when executed resulting in a performance of:

Claim 12 of the Patent (Patent No 7,593,791 B2), lines 1-4, and 10-11 recites as follows:

(A computer program having program code executable by a computer, the program code, when executed by the computer causing the computer to perform the steps of: determining a breakpoint in the at least one program;

assigning the global variable address information which is present in at least one memory device (Assigning the address information by means of determining a global variable) -emphasis added):

providing at least one stimulation function;

Claim 12 of the Patent (Patent No 7,593,791 B2), lines 12-15 recites as follows:

(loading the memory device the address information of the global variable by at least one load; and replacing the address information of the global variable with one of the store instruction and the load instruction (these executable steps are done through stimulating the external function to activate the breakpoint for controlling the sequence of operation in a control unit –See specification (Patent No 7,593,791 B2) Col 3:25-36.

and accessing the at least one global variable via at least one software breakpoint

Claim 12 of the Patent (Patent No 7,593,791 B2), lines 4, and 10-15 recites as follows:

(determining a breakpoint in the at least one program;
assigning the global variable address information which is present in at least one memory device; loading the memory device the address information of the global variable by at least one load; and replacing the address information of the global variable with one of the store instruction and the load instruction)
Thus it would have been obvious to one ordinary skill in the art to recognize that after determining the break point of the function at first then the global variable address information is accessed to assign and replace address information –*See* remaining limitations of claim 1 of the patent.

<u>Per claim 30:</u>

The computer program product as recited in Claim 29, *wherein a code of the product is executed on a computer*

Claim 12 of the Patent (Patent No 7,593,791 B2), lines 1-4 recites as follows:
(A computer program having program code executable by a computer, the program code, when executed by the computer causing the computer to perform the steps of).

Claim Rejections - 35 USC § 102

13. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

14. Claims 16-30 are rejected under 35 U.S.C. 102(b) as being unpatentable by Iain Haswell-Smith (US Patent No. 4,866,665 –IDS of record).

Per claim 16:

Haswell-Smith discloses—

A method for stimulating functions for controlling operating sequences, the functions accessing at least one global variable of at least one program for control (Abstract –software break points are defined at desired addresses (global variable) by replacing user instructions at such addresses with call instructions (stimulating function) that call a monitor routine which handles software break points –emphasis added), *the method comprising:*
providing at least one stimulation function (Col 2:61-67 –Most monitors that are executed (provided) in a system using a microprocessor whose instruction set includes a trap instruction have a "singe step command" which easily enables a user program to be executed one instruction at a time The sequence of events of a monitor single step command are as follows The monitor (stimulating function) examines the op code in the memory whose address is given by the value of the user program counter –emphasis added); and
accessing the at least one global variable via at least one software breakpoint

(Abstract –software break points are defined at desired addresses by replacing user instructions at such addresses (instruction address values are global) with call instructions that call a monitor routine which handles software break points. The argument of the call instruction is made equal to its op code. The portion of the monitor that handles software break points is entered at an address (by accessing at the point of halt or break) equal to the op code of the call instruction inserted at the break point address –emphasis added).

Per claim 17:

Haswell-Smith discloses—

wherein the functions are stimulated within the at least one program during a runtime of the at least one program (Col 1:38-44 –To halt the execution of a user program (run time of the program to be monitored), a software break point is set at the particular location in memory where a halt is desired. The software break point causes the microprocessor to stop executing the user program, save all the current values of its registers in a certain area of memory reserved for exclusive use by the monitor, and then start executing the monitor program –emphasis added).

Per claim 18:

Haswell-Smith discloses—

wherein the functions are stimulated in real time within a runtime system (Col 6:38-41 – After the system is powered up (start executing controlling function), the user wishing to operate the software monitor will define various software break points in the user program stored in section 46 (break points are already defined) of the memory map 45; Col 7:36-67, Col 8:1-3 - When the monitor program is entered via a software break point, the monitor saves all the microprocessor register values in memory (the second form of stack). This means a monitor command such as "display registers" actually reads this area of memory and displays the contents thereof. If the monitor "run" command is executed to run a user program the monitor reloads all the saved register values from memory back into the processor, including the previously saved program counter value, and then restarts execution of the user program from the reloaded program counter value. The SPV120 monitor actually only uses one of the registers of the internal stack within the microprocessor This swapping back and forth of all the microprocessor register values between the microprocessor and memory when a user program or a monitor program is halted or started is known technically as a "context switch.")

The monitor then executes the instructions of the break point handling portion of the monitor routine beginning with address E020, and replaces the trap instruction at location 2002 with the user instruction that was originally stored at that address and displays the saved contents of all of the microprocessor registers corresponding to break point address 2002. The monitor then returns to the beginning of its main section 47A and awaits the next monitor command. Appendix A attached hereto contains a list of monitor commands for the monitor contained in EPROM 11 and is typical of monitor command sets.

If the next monitor command is a monitor run (i.e., "execute program code") command, the microprocessor jumps to location 2003 of the user program, and continues to execute the user program until the next break point is encountered, and from that point on repeats the above-described procedure (Thus, monitor program (stimulating program) runs in between the consecutive break points, and after executing various monitor routines it returns the execution control back to the user program (controlling program) which in turn returned back to monitor program. In other words both monitor program and user program executed interactively by means of context switching, which further means monitor program runs within real time execution of user program) –emphasis added).

Per claim 19:

Haswell-Smith discloses—

assigning the at least one global variable a first piece of data information (Col 2:12-16 –the monitor "set break point" command replaces an instruction in the user program with a trap instruction (first piece of data information) at the address (global variable) of each desired software break point and temporarily stores the replaced user program instructions);
replacing the first piece of data information by a second piece of data information corresponding to new stimulation values (Col 2:19-23 –a two word branch instruction that calls the monitor subroutine which handles software break points and replaces every trap instruction with the user instruction (second piece of data information) that earlier was replaced by the trap instruction).

Per claim 20:

Haswell-Smith discloses—

stimulating the functions by an internal bypass (Col 4:38-43 –when initially commanded to run, replaces the user program instruction at the desired break point address with the op code of a call instruction and also replaces the subsequent instruction with the argument or address to be called by the call instruction (internal bypass) –emphasis added).

Per claim 21:

Haswell-Smith discloses—

the software breakpoint is accomplished by assigning address information to the at least one global variable (Col 2:12-16 –the monitor "set break point" command replaces an instruction in the user program with a trap instruction (first piece of data information) at the address (global variable) of each desired software break point and temporarily stores the replaced user program instructions), *the address information is loaded from a memory means by a load instruction* (Col 9:60-61 –all the user register values are loaded from the memory stack into the microprocessor), and
the address information of the at least one global variable of the load instruction is replaced (Col 6:61-64 –Execution of the (monitor run) command causes the instruction at the address of each break point to be replaced by a trap instruction, and also causes each replaced user instruction to be temporarily stored).

Per claim 22:

Haswell-Smith discloses—

replacing the address information of the at least one global variable by address information of a pointer variable (Col 6:43-47 –A trap instruction is a single word instruction that contains a fixed vector. When the trap instruction is encountered by microprocessor 9, it jumps to the address pointed to by the trap instruction (use of pointer variable) –emphasis added).

Per claim 23:

Haswell-Smith discloses—

determining an initial address of the at least one stimulation function from the address information (Col 6:10-14 –The typical software monitor program (stimulation function) is included in section 47 of <u>memory map 45, beginning with the address E000</u>. The software monitor, includes a main part 47A and a "break point handler" section 47B that begins at address E020 –emphasis added).

<u>Per claim 24:</u>

Haswell-Smith discloses—

replacing functions for control of operating sequences by replacing the address information by additional functions (Col 3:14-18 -The trap instruction is executed, causing the user program to be halted and the monitor program to execute. The <u>monitor program saves all the processor register values</u> and replaces (through run command which is additional function) the trap instruction with <u>the original saved user instruction</u> (function for control of operating sequences) –emphasis added).

<u>Per claim 25:</u>

Haswell-Smith discloses—

the software breakpoint is accomplished by addressing the at least one global variable via a store instruction (Abstract –<u>software break points are defined</u> at desired addresses <u>by replacing user instructions at such addresses with call instructions</u> that call a monitor routine which handles software break points –emphasis added), and
the store instruction is manipulated onto the at least one global variable by replacing the store instruction by a jump instruction (Col 4:50-54 –The monitor

also saves the address of the first break point, i.e., the current contents of the program counter, and the other registers of the microprocessor on a stack, and jumps (using jump instruction) to the portion of the monitor routine that handles break points).

Per claim 26:

Haswell-Smith discloses—

wherein the functions for controlling the operating sequences by replacing the store instruction by the jump instruction are replaced by additional functions (Col 4:50-54 –The monitor also saves the address of the first break point, i.e., the current contents of the program counter, and the other registers of the microprocessor on a stack, and jumps (using jump instruction) to the portion of the monitor routine that handles break points).

Per claim 27:

Haswell-Smith discloses—

A device for stimulating functions for controlling operating sequences, the functions accessing at least one global variable of at least one program for control (Abstract –software break points are defined at desired addresses (global variable) by replacing user instructions at such addresses with call instructions (stimulating function) that call a monitor routine which handles software break points –emphasis added), comprising:
a first stimulation arrangement for activating at least one stimulation function (Col 2:61-67 –Most monitors that are executed (provided) in a system using a microprocessor whose instruction set includes a trap instruction have a "singe step command" which easily enables a user program to be executed one

instruction at a time The sequence of events of a monitor single step command are as follows The monitor (stimulating function) examines the op code in the memory whose address is given by the value of the user program counter –emphasis added); and

a control arrangement for generating at least one software breakpoint (Col 6:38-41 -After the system is powered up, the user wishing to operate the software monitor(stimulating function) will define various software break points in the user program stored in section 46 of the memory map 45), *the at least one stimulation function accessing the at least one global variable via the software breakpoint* (Abstract –software break points are defined at desired addresses by replacing user instructions at such addresses (instruction address values are global) with call instructions that call a monitor routine which handles software break points. The argument of the call instruction is made equal to its op code. The portion of the monitor that handles software break points is entered at an address (by accessing at the point of halt or break) equal to the op code of the call instruction inserted at the break point address –emphasis added).

Per claim 28:

Haswell-Smith discloses—

A control unit, comprising:

a device for stimulating functions for control of operating sequences, the functions accessing at least one global variable of at least one program for control (Abstract –software break points are defined at desired addresses (global variable) by replacing user instructions at such addresses with call instructions (stimulating function) that call a monitor routine which handles software break points –emphasis added), the device including:

a first stimulation arrangement for activating at least one stimulation function (Col 3:9-15 –The monitor then executes a run command (as part of the single step command). The user program instruction is executed and the program counter is incremented to the address value of memory that contains the trap instruction. The trap instruction is executed, causing the user program to be halted and the monitor program to execute),

and

a control arrangement for generating at least one software breakpoint (Col 6:38-41 -After the system is powered up, the user wishing to operate the software monitor(stimulating function) will define various software break points in the user program stored in section 46 of the memory map 45), *the at least one stimulation function accessing the at least one global variable via the software breakpoint* (Abstract –software break points are defined at desired addresses by replacing user instructions at such addresses (instruction address values are global) with call instructions that call a monitor routine which handles software break points. The argument of the call instruction is made equal to its op code. The portion of the monitor that handles software break points is entered at an address (by accessing at the point of halt or break) equal to the op code of the call instruction inserted at the break point address –emphasis added).

Per claim 29:

Haswell-Smith discloses—

A computer program product *for stimulating functions for controlling operating sequences, the functions accessing at least one global variable of at least one*
program for control (Abstract –software break points are defined at desired addresses (global variable) by replacing user instructions at such addresses with call instructions (stimulating function) that call a monitor routine which handles software break points –emphasis added), *the product when executed resulting in a performance of: providing at least one stimulation function* (Col 3:9-

15 –The monitor then executes a run command (as part of the single step command). The user program instruction is executed and the program counter is incremented to the address value of memory that contains the trap instruction. The trap instruction is executed, causing the user program to be halted and the monitor program to execute); and *accessing the at least one global variable via at least one software breakpoint* (Abstract –software break points are defined at desired addresses by replacing user instructions at such addresses (instruction address values are global) with call instructions that call a monitor routine which handles software break points. The argument of the call instruction is made equal to its op code. The portion of the monitor that handles software break points is entered at an address (by accessing at the point of halt or break) equal to the op code of the call instruction inserted at the break point address –emphasis added).

Per claim 30:

Haswell-Smith discloses—

a code of the product is executed on a computer (Col 1:14-16 –Programs called software monitors, or simply "monitors", commonly are used in computer systems to enable engineers to "debug" or analyze software).

Conclusion

15. The prior art made of record and have yet relied upon is considered pertinent to applicant's disclosure.

(A). Haag et al. (US Patent Application Publication No. 2002/0198638 A1), discloses —A method and an apparatus for monitoring the control of operational sequences in a vehicle is described, in which context control functions are executed in a control unit, and monitoring functions that monitor the control functions are also executed –*See* Abstract.

Haag further discloses —At least one monitoring region may be created from the control functions of the first functional plane, in such a manner that selectable functions which form a sequence that is constant with respect to the run time of the control of the operational sequences are combined in the at least one monitoring region –*See* Summery.

(B). Beuten et al. (US Patent Application Publication No. 2002/0073400 A1), discloses —A method for monitoring the execution of a program executable on at least one microprocessor of a micro controller using a debug logic of the micro controller, with an exception, particularly an interrupt of the program execution, being triggered by the debug logic upon access to a specific address range during the program execution time. In order to provide monitoring of the execution of a program, executable on a microprocessor, for faults, which is particularly reliable yet conserves resources as much as possible, it is suggested that the debug logic be configured by the microprocessor and that an exception routine be executed by the debug logic after an exception has been triggered during the program execution time. The debug logic is advantageously configured during the startup of the micro controller. Preferably, the micro controller is reset and started up again and the monitored program is initialized during the execution of the exception routine –*See* Abstract.

(C). Yutaka Haga (US Patent Publication No. 6,715,140 B1), discloses — A program trace apparatus intercepts control during execution thereof by rewriting of an instruction sequence of a program and traces an execution state of the original program. The program trace apparatus includes a judging section for judging whether or not an arbitrary instruction which is to be rewritten to intercept the control is longer than a control transfer instruction which is written over the arbitrary instruction to intercept the control, and an identification section for identifying a trace point by storing an index data for identifying a rewritten location in one or more surplus bytes of the arbitrary instruction if the judging section judges that the arbitrary instruction is longer than the control transfer instruction and the one or more surplus bytes exist in the arbitrary instruction –*See* Abstract.

16. Any inquiry concerning this communication or earlier communications from the examiner should be directed to ZIAUL CHOWDHURY whose telephone number is (571)270-7750. The examiner can normally be reached on Monday Thru Friday, 7:30AM To 9:00PM, Alternet Friday, Eastern Time.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, TUAN Q. DAM can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/ZIAUL CHOWDHURY/
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192